

# **Weighting Procedures and the Cluster Forming Algorithm for Delete-k Jackknife Variance Estimation for Institutional Surveys**

**Jiahe Qian**

**Weighting Procedures and the Cluster Forming Algorithm for Delete-k Jackknife Variance  
Estimation for Institutional Surveys**

Jiahe Qian  
ETS, Princeton, NJ

June 2006

As part of its educational and social mission and in fulfilling the organization's nonprofit charter and bylaws, ETS has and continues to learn from and also to lead research that furthers educational and measurement research to advance quality and equity in education and assessment for all users of the organization's products and services.

ETS Research Reports provide preliminary and limited dissemination of ETS research prior to publication. To obtain a PDF or a print copy of a report, please visit:

<http://www.ets.org/research/contact.html>

Copyright © 2006 by Educational Testing Service. All rights reserved.

ETS and the ETS logo are registered trademarks of  
Educational Testing Service (ETS).



## **Abstract**

Weighting and variance estimation are two statistical issues involved in survey data analysis for large-scale assessment programs such as the Higher Education Information and Communication Technology (ICT) Literacy Assessment. Because survey data are always acquired by probability sampling, to draw unbiased or almost unbiased inferences for the populations, weights are required in making use of estimators such as a Horvitz-Thompson type. Variance estimation provides the basis for reporting errors. The weighting procedure generates weights based on statistical principles that are consistent with the sampling design. The estimation of the variance from survey data uses the delete- $k$  jackknife resampling replicate (JRR) approach, which can be adapted for variant institutional sampling designs and for dissimilarity in institute conditions. To form clusters of  $k$  cases, a merge-dilute algorithm is proposed. The algorithm merges the cases of different groups into a queue and then allocates the cases of the queue to form homogeneous clusters of required sizes. The new algorithm is applied to the ICT sample from an institute taking the 2004 fall trial assessment.

Key words: Horvitz-Thompson estimator, weight adjustment, variance estimation, merge-dilute algorithm

### **Acknowledgements**

This research was supported by the Higher Education ICT Literacy Assessment program at ETS. The author thanks Shelby Haberman, Dan Eignor, David Williamson, and Frank Jenkins as well as ICT research scientists for their suggestions and comments, especially for Dan Eignor's substantive help in editorial issues. The author also would like to thank Bruce Kaplan for his planning work on the project and Sailesh Vezzu for his work on SAS programming.

## Introduction

This paper lays out the weighting procedure for institutional surveys and the cluster-forming algorithm for the delete- $k$  jackknife resampling replicate (JRR) approach and then applies these procedures to the data from one institution that took the Higher Education Information and Communication Technology (ICT) Literacy Assessment (Jenkins & Qian, 2005; Williamson, Katz, & Redman, 2005). This example is a large-scale institutional assessment, providing report cards for institutes and their subpopulations. To make inferences for the populations of interest, institutional surveys must collect data by probability sampling from the institutes that participate.

An institutional survey usually attempts to sample cases with approximate equal chances of selection. However, due to special interest in domains of study and due to variations in institute conditions, the cases are usually included in a sample with unequal probabilities. Therefore, to achieve unbiased estimates of statistics such as totals, means, and percentages, weights need to be applied in the Horvitz-Thompson estimators (Cochran, 1977; Kish, 1965). Moreover, weights also need to be adjusted for nonresponse and poststratification.

The variances for statistics of interest are estimated by using the delete- $k$  JRR approach (Rust, 1985; Shao & Tu, 1995; Shao & Wu, 1989; Wolter, 1985). The approach used in the paper is an extension of the method used in operational National Assessment of Educational Progress (NAEP; Allen, Donoghue, & Schoeps, 2001; Rust, 2004). The JRR approach approximates the distribution of the estimates by the empirical distribution of replicates and estimates the sampling variance by the variability among the replicates. Moreover, the delete- $k$  JRR approach provides a balance between the number of replicates and the sizes of clusters.

Section 1 of this paper describes the development of a weighting procedure to derive weights to perform unbiased estimation by the Horvitz-Thompson estimators. The procedure involves adjustments for nonresponse and poststratification.

Section 2 introduces the delete- $k$  JRR approach. Instead of dropping one case, the delete- $k$  JRR process drops a cluster of  $k$  cases in each replicate. A new methodology, the merge-dilute algorithm, is proposed to form the clusters for delete- $k$  JRR variance estimation. The algorithm efficiently allocates different groups of cases into an evenly sorted queue that allows a flexible clustering strategy for the delete- $k$  JRR procedure.

In Section 3, the procedure and algorithm developed in Sections 1 and 2 are applied to the sample from an institute that took the 2004 fall ICT trial assessment. Several sets of variance estimate results under different cluster forming schemes are compared. The results reveal the validity of the proposed variance estimation procedure.

As concluded in Section 4, results in this report show the efficacy and applicability of the proposed weighting procedure and delete- $k$  JRR approach. The framework can be adapted to similar situations involving other large-scale educational assessments.

## 1. Weighting Procedure for Institutional Surveys

To obtain unbiased or less biased estimation from survey data, weights need to be used in estimating statistics for reporting. The weighting procedure consists of three steps: (a) compute base weights for cases that have participated in the assessment, (b) adjust for nonresponse, and (c) conduct poststratification or raking.<sup>1</sup>

### *1.1 The Horvitz-Thompson Estimator*

Let  $\pi_i$  be the probability that case  $i$  is included in the sample and  $y_i$  be the value of the variable of interest, measured from case  $i$ . When a sample is selected without replacement by probability sampling, the Horvitz-Thompson estimator of the population total ( $\hat{Y}$ ) is

$$\hat{Y} = \sum_{i \in R} \frac{y_i}{\pi_i},$$

where  $R$  is the set of sampled cases of size  $n$ . The estimator  $\hat{Y}$  is unbiased (Cochran, 1977; Kish, 1992). Let case weight  $w_i$  equal the inverse of the probability of selection. The target statistic of mean or proportion can be estimated by a ratio estimator. When the mean is the target, its estimate ( $\bar{y}$ ) is

$$\bar{y} = \frac{\sum_{i \in R} w_i \cdot y_i}{\sum_{i \in R} w_i}.$$

Nevertheless, the ratio estimator  $\bar{y}$  of the mean is biased with an order of  $O(1/n)$  (Cochran, 1977).

A typical institute sample design, like that employed with ICT, would involve stratified simple random sampling. Define stratum weight  $W_h = N_h / N$  ( $h = 1, \dots, L$ ), where  $N_h$  is the size

of stratum  $h$  (e.g., freshman vs. junior), and  $N$  is the size of the population. Let  $y_{hk}$  ( $k = 1, 2, \dots, n_h$ ) be the value measured from case  $k$  in stratum  $h$ , and let  $w_{hk}$  be its corresponding weight.

The mean estimate for stratum  $h$  ( $\bar{y}_h$ ) is

$$\bar{y}_h = \frac{\sum_k w_{hk} \cdot y_{hk}}{\sum_k w_{hk}}.$$

Then the estimator  $\bar{y}$  can be expressed as

$$\bar{y} = \sum_{h=1}^L W_h \cdot \bar{y}_h.$$

Let case weights be normalized to stratum size:  $N_h = \sum_k w_{hk}$ . (For the descriptions of normalization of weights, see Section 1.4.) Then the estimate of the mean is

$$\bar{y} = \sum_{h=1}^L W_h \cdot \bar{y}_h = \frac{\sum w_i \cdot y_i}{\sum w_i}.$$

### ***1.2 Case Base Weights***

Let  $\pi_{h g k}$  be the inclusion probability for case  $k$  in stratum  $h$  and subpopulation  $g$ . Then the basic weight for case  $k$  is

$$w_{\mathcal{B}, h g k} = \frac{1}{\pi_{h g k}}.$$

The symbol  $\mathcal{B}$  in the subscript stands for base weights.

### ***1.3 Adjustment for Nonresponse for Base Weights***

Two types of nonresponse occur in educational surveys: case nonresponse and item nonresponse. The case nonresponse occurs when a sampled individual does not respond to the request to be assessed. The causes of the case nonresponse could be noncontact or noncooperation. The item nonresponse refers to the failure to give answers to particular items on a test. Both types of nonresponse can be important sources of error in assessments, but only case



nonresponse is considered in the weighting process, whereas the item nonresponse is handled by the scaling process of item response theory models.

Case nonresponse will cause a systematic difference between sample-based estimates of population statistics and their true values when respondents differ from nonrespondents on their ability to be measured. However, if respondents and nonrespondents are interchangeable on some characteristics for certain subgroups, weight adjustment can be applied to account for the case nonresponse.

In the weight adjustment for case nonresponse, adjustment classes first need to be formed by the demographic variables of interest, such as gender and ethnicity. Occasionally the adjustment classes are the same as the subpopulations of interest within each stratum. Then, the base weights are multiplied by a factor in each adjustment class to make the assessed counts equal to the sampled counts by design. This adjustment is based on the assumption that the responses missing within each adjustment class are at random. Let nonresponse factor  $f_{\mathcal{A},hg}$  in stratum  $h$  and subpopulation  $g$  equal the inverse of the response rate. Let the symbol  $\mathcal{A}$  in the subscript stand for adjustment. The formula for adjustment is

$$w_{\mathcal{A},hg} = w_{\mathcal{B},hg} \cdot f_{\mathcal{A},hg}.$$

The symbol  $w_{\mathcal{B},hg}$  implies that the base weights are the same for all the cases in stratum  $h$  and subpopulation  $g$ . Note that  $f_{\mathcal{A},hg} \geq 1$ .

#### ***1.4 Adjustment of Weights by Poststratification and Raking***

After nonresponse adjustment, some variables could show considerable gaps between a weighted sample distribution and its corresponding population distribution. Such gaps are revealed in the corresponding cells that are cross-classified by the variables. The inconsistency between sample and population arises from sampling fluctuation, response errors, or frame defects.<sup>2</sup> Poststratification and raking can be used to correct for these known gaps between the sample distribution and the population distribution, to improve the precision of the survey estimates by reducing their mean squared error, and to enhance the comparability of the survey data under study with data from other surveys.

Poststratification adjustment matches the weighted sample cell counts to the population cell counts by applying a proportional adjustment to the weights in each cell across the

contingency table (Kish, 1965). Sometimes, however, the sample can be spread too thinly across the cells on the table. Therefore, poststratification would produce extreme weights in the cells with few cases and cause large weighting effects. To avoid such flaws, raking is used to control marginal distributions for the variables of interest.

A raking procedure iteratively adjusts the case weights in the sample to make the weighted marginal distributions of the sample agree with the marginal distributions of the population on specified demographic variables (Deming, 1943). The algorithm used in raking is called the Deming-Stephan algorithm (Deming & Stephan, 1940; Haberman, 1979).

The process of poststratification consists of two main steps. First, the case weight is adjusted by multiplying it by a poststratification factor,  $f_c^r$ , where  $c$  is the cell index in poststratification. Let gender be involved in poststratification and  $w_{h g k}$  be the case weight for case  $k$  in stratum  $h$  and subpopulation  $g$ . For a case in cell  $c = 1$ , then  $w_{h g k} = w_{A, h g k} * f_1^r$ , and in cell  $c = 2$ , then  $w_{h g k} = w_{A, h g k} * f_2^r$ . Second, the sum of the case weights needs to be normalized. The weight normalization refers to the adjustment of weights by multiplying a constant with each weight in a sample or subsample so that the sum of weights is equal to a defined size (e.g., population/subpopulation size, sample/subsample size), or one. If the sum of weights is normalized to one, a mean can be estimated by the weighted sum. The sum of case weights within each stratum is normalized to the stratum population total,

$$\sum_g \sum_k w_{h g k} = N_{h..} ,$$

where  $N_{h..}$  is the population size of stratum  $h$  ( $h = 1, \dots, L$ ) and  $L$  is the total number of strata.

To reduce weighting effects, the weight adjustment process usually includes a step of weight trimming. The trimming process truncates extreme weights caused by unequal probability sampling or by nonresponse and poststratification adjustment. It reduces variation caused by extremely large weights but introduces some bias in estimates. The process usually employs the criterion of minimum mean squared error (Potter, 1990). Weight trimming adds complexity to the weighting procedure. Because institute programs attempt to select cases with equal probabilities, their samples usually do not yield extreme case weights. Therefore, such programs, including the 2004 fall trial ICT assessment, likely do not need the weight trimming process.

## **2. Estimation of Sampling Variance**

The delete- $k$  JRR approach is used to estimate the variances of statistics in reporting because it provides a balance between the number of JRR replicates and the sizes of clusters. This flexibility enables testing programs to comply with diverse requirements from varied institutes and variations in sampling designs. The implementation of the delete- $k$  JRR approach mainly consists of two steps: forming the clusters of sampled cases and estimating the jackknifed variance. In the process of variance estimation, by applying a newly proposed merge-dilute algorithm, the cases are first formed into clusters of size  $k$ . Then, the process computes a replicate mean estimate from the sample by dropping one cluster. When all the replicates are calculated, the variance of the mean estimate is estimated by the variability among the JRR replicate estimates. Section 2.1 describes the cluster forming scheme. Section 2.2 discusses the proposed algorithm. Section 2.3 describes the jackknifing process.

### ***2.1 Forming Student Clusters for the Delete- $k$ JRR Approach***

To form clusters, the cases within each stratum are partitioned into groups by their demographic characteristics such as gender and ethnicity. Because the demographic variables correlate with the variables measured, to estimate variation due to sampling, the empirical rule in JRR is to form clusters that are homogeneous to each other (Allen et al., 2001). Therefore, the rule for forming clusters is to evenly allocate cases with different demographic characteristics into each cluster. If possible, a cluster should be formed by assigning a similar number of cases from each group. When a group runs out of cases, a cluster is then formed by assigning more cases from the groups that are not used up. In some extreme situations, a cluster will contain cases from the same group. For example, assume cases are partitioned into four groups, gender by minority status, and the cluster size is 4 cases. If the size of one group is larger than 75% of the sample size, some clusters will contain multiple cases from the same group.

### ***2.2 The Merge-Dilute Algorithm***

In this study, a merge-dilute algorithm is designed to form the clusters. It merges several small groups into one group that is called a *queue*, and merges the cases by controlling the interval of cases from the same demographic group and the length of the queue of the cases from the same demographic group. A SAS macro program, in Appendix A, implements the merge-dilute algorithm.

*The merge-dilute algorithm for two groups.* First, consider a simple situation: applying the merge-dilute algorithm to merge two demographic groups. Let the cases in each group be randomly sorted. Let integers  $s_1$  and  $s_2$  ( $s_1 \leq s_2$ ) be the sizes of the first and second groups. Let  $k$  and  $s_0$  be integers and  $s_2 = ks_1 + s_0$  ( $s_0 < s_1$ ). Then  $s_2 + s_1 = (k+1)s_1 + s_0$ . Let  $r = s_1 - s_0$ . Then

$$s_1 = r + s_0,$$

$$s_2 = kr + (k+1)s_0,$$

and

$$s_2 + s_1 = (k+1)r + (k+2)s_0.$$

Let  $d_1$  ( $d_1 = 1, 2, \dots, s_1$ ) be the original index of the cases in the first group. Then each case in the first group is assigned a new index:

$$d^* = \begin{cases} (k+1) \cdot d_1 & , \quad d_1 \leq r, \\ (k+2) \cdot d_1 - r & , \quad d_1 > r. \end{cases}$$

Let  $[c]_k$  denote the group of numbers congruent to  $c$  modulo  $k$ .<sup>3</sup> The symbol  $[c]_k$ , particularly in this paper, is also used to identify with the corresponding remainders, and the possible results of  $[c]_k$  are  $0, 1, \dots, k-1$ . In the example of  $13 = 5 \cdot 2 + 3$ ,  $[13]_5 = [3]_5 = 3$ . For  $27 = 7 \cdot 3 + 6$ ,  $[27]_7 = [6]_7 = 6$ . Note that  $[0]_k = 0$ .

For the second group, let the original index of the cases be  $d_2$  ( $d_2 = 1, 2, \dots, s_2$ ). Let  $\alpha = d_2 - 1$  and  $\beta = \alpha - kr$ . Then each case in the second group is assigned a new index:

$$d^* = \begin{cases} (k+1) \cdot \frac{\alpha - [\alpha]_k}{k} + [\alpha]_k + 1 & , \quad d_2 \leq kr, \\ (k+1) \cdot r + (k+2) \cdot \frac{\beta - [\beta]_{(k+1)}}{k+1} + [\beta]_{(k+1)} + 1 & , \quad d_2 > kr. \end{cases}$$

Generated by the above formulas, the new index  $d^*$  is used as the index of the queue merged from the two original groups under consideration. Then the cases in the queue are sorted by the index  $d^*$ .

As an illustrative example, assume the sample sizes of two groups to be merged are 5 and 13. The set of the original index  $d_1$  for the first group is  $\{1, 2, 3, 4, 5\}$ , and the set of  $d_2$  for the second group is  $\{1, 2, \dots, 13\}$ . Making use of the equations  $s_2 = ks_1 + s_0$  and  $r = s_1 - s_0$ , then  $13 = 2 \cdot 5 + 3$  and  $2 = 5 - 3$ . Applying the formula for the new index, the set of  $d^*$  for the cases in the first group is  $\{3, 6, 10, 14, 18\}$ , and the set of  $d^*$  for the cases in the second group is  $\{1, 2, 4, 5, 7, 8, 9, 11, 12, 13, 15, 16, 17\}$ .

*Merge G groups by the algorithm.* After the previous case of merging two groups has been considered, the algorithm can be generalized to merging  $G (>2)$  groups. Assume the cases within each group are randomly sorted. The groups are sorted by their sample sizes:

$s_1 \leq s_2 \leq \dots \leq s_G$ . The first step applies the algorithm to merge two groups of the two smallest sizes ( $s_1$  and  $s_2$ ) and create the new index for the merged group. Then, for  $G - 1$  groups, the groups are sorted by their sample sizes again:  $s_1^* \leq s_2^* \leq \dots \leq s_{G-1}^*$  and the algorithm is applied to merge two groups with sizes of  $s_1^*$  and  $s_2^*$ . The procedure is repeated until all groups are merged into one queue. It takes  $G - 1$  steps to accomplish the process of merging  $G$  groups.

*Assign cluster index.* After all  $G$  groups are merged into one new queue and the cases are sorted by the new index, the cases are partitioned into clusters by assigning a cluster index to each case. Let  $d^{**}$  be the new index of the merged group and  $m$  be the cluster size for the delete- $k$  JRR approach. Note that  $k = m$ . Then the cluster index is defined as

$$j = \frac{d^{**} - \lfloor d^{**} \rfloor_m}{m} + 1.$$

The largest cluster index ( $J$ ) equals  $m^{-1} (n - \lfloor n \rfloor_m) + 1$ , where  $n$  is the sample size. To define replicates, let  $R_{(j)}$  be the replicate set of the sample by dropping cases with the cluster index  $j$  ( $=1, 2, \dots, J$ ). Note that if the size of the last cluster is too small, the cases in the last several clusters may need to be adjusted according to specific situations.

*Property of the merge-dilute algorithm.* Let  $a$  and  $b$  be two cases in a queue merged from two groups and their case indices be  $d_a$  and  $d_b$ . To analyze the property of the merge-dilute algorithm, define the distance between  $a$  and  $b$  as

$$d(a,b) = |d_a - d_b|.$$

The distance between  $a$  and  $b$  is called *neighboring distance* if  $a$  and  $b$  are from the same group and if the cases between  $a$  and  $b$  are all from the other group. According to the formula for the case index of the merge-dilute algorithm, the neighboring distance of two cases from the first group is either  $k + 1$  or  $k + 2$ , and the neighboring distance of two cases from the second group is either 1 or 2. Therefore, the maximum neighboring distance of two cases in the merged group is  $k + 2$ . A queue of cases is called a *continual queue* if it only comprises cases from the same group and with sequential case indexes. According to the algorithm, the length of the continual queue for cases from the first group is 1, and the length of the continual queue for cases from the second group is either  $k$  or  $k + 1$ . For a group merged from two demographic groups, the length of the maximum continual queue is equal to or less than  $k + 1$ .

In the example in Section 2.2.1, the neighboring distance for cases from the first group is either 3 or 4, and the neighboring distance for cases from the second group is either 1 or 2. The length of the continual queue for cases from the first group is 1, and the length of the continual queue for cases from the second group is either 2 or 3. Thus, the maximum neighboring distance equals 4 and the length of the maximum continual queue is 3.

In general, the maximum neighboring distance and length of the maximum continual queue are not equal for a merged queue. As an extreme example, for two groups of sizes  $s_1$  and  $s_2$  ( $s_1 \leq s_2$ ), let all the cases in the first group be put in front of the cases in the second group. Then, the maximum neighboring distance equals 1, and the length of the maximum continual queue is  $s_2$ .

The following discussion will show that the algorithm yields an evenly diluted queue when merging two groups. The criterion of an even merger is whether the occurrence rate for each case from the same group on each sequential queue is the same or close. The *sequential queue* is defined as the set of the cases with sequential case indices. For a special situation  $s_0 = 0$  (i.e.,  $r = s_1$ ), each neighboring distance of two cases from the first group equals  $k + 1$ . Moreover, each continual queue of cases from the second group is  $k$ . For every sequential queue of length  $k + 1$ , when  $s_0 = 0$ , the occurrence of cases from the first group is  $(k + 1)^{-1}$ . Therefore, the

occurrence of cases from the second group is  $(k+1)^{-1} \cdot k$ . The occurrence rate for each case from the same group is the same. Therefore, the queue is evenly merged.

For a general situation,  $s_0 > 0$ , the neighboring distance of two cases from the first group equals either  $k+1$  or  $k+2$ ; the continual queue of cases from the second group is  $k$  or  $k+1$ . Thus, the merged queue has two distinct neighboring distances for the cases from the first group, and the difference between two distinct neighboring distances is one. Additionally, the merged queue has two distinct continual queues for the cases from the second group, and the difference between two distinct continual queues is also one. When  $s_0 > 0$ , no plan can yield a merged queue with identical neighboring distances and identical continual queues for cases from the same group. Consequently, it is impossible to obtain an identical occurrence rate on each sequential queue with a fixed length. For a strategy that yields a queue with three or more distinct neighboring distances for cases from the first group, the difference between two distinct neighboring distances would be larger than one. Analogously, for the continual queues yielded by the plan, the difference between two distinct continual queues also would be larger than one. Such a plan would not be better than the strategy defined by the merge-dilute algorithm.

Consider the occurrence rate for a queue merged by using the merge-dilute algorithm when  $s_0 > 0$ . For the cases with case indexes less than  $r+k+1$ , the occurrence rate of the cases from the first group is  $(k+1)^{-1}$  for each sequential queue of length  $k+1$ . For the cases with the case index larger than  $r$ , the occurrence rate of the cases from the first group is  $(k+2)^{-1}$  for each sequential queue of length  $k+2$ . Although they are not identical, the two occurrence rates are as close as possible.

From the discussion above, for a general condition ( $s_0 > 0$ ), the merge-dilute algorithm yields a merged group with the maximum neighboring distance of  $k+2$  and the maximum continual queue equal to or less than  $k+1$ . By applying the pigeonhole principle (Knuth, 1968; Lovasz, Pelikan, & Vesztergombi, 2003), no strategy to merge two groups will satisfy the following two conditions simultaneously: the maximum neighboring distance is less than  $k+2$  and the length of the maximum continual queue is less than  $k+1$ . Therefore, the algorithm is an optimal strategy for yielding an evenly diluted queue from two groups.

Because the property of the algorithm does not involve the variable cluster size, it allows choosing the optimal number of replicates based on the requirement of the delete- $k$  JRR approach. Such flexibility provides the convenience of using one software package to analyze the samples from different designs under different institute conditions.

For a sample of  $G$  groups, it takes  $G - 1$  steps to form a merged queue by the proposed algorithm. In the first step, the two smallest groups are merged by the algorithm. Then, including the merged group,  $G - 1$  groups remain. After this procedure is repeated  $G - 1$  times, one queue is obtained that is merged from these  $G$  groups. Let the maximum neighboring distance of two cases in each step be  $k'_1 + 2, k'_2 + 2, \dots, k'_{G-1} + 2$ . Let  $k^* = \max \{k'_1, k'_2, \dots, k'_{G-1}\}$ . Then, the maximum neighboring distance of two cases in the group is  $k^* + 2$ . Moreover, the length of the maximum continual queue is equal to or less than  $k^* + 1$ . By the same reasoning, for two groups, the procedure yields an evenly diluted queue from  $G$  groups.

### 2.3 Computation of the Variance of Mean Estimates

*Variance estimation by the delete- $k$  JRR approach.* To calculate replicates, the JRR process repeatedly drops a cluster of cases from the sample and computes the replicate estimates, which are called the pseudo-values of estimates. The  $j$ th replicate estimate  $\bar{y}_{(j)}$  equals

$$\bar{y}_{(j)} = \frac{\sum_{i \in R_{(j)}} w_i \cdot y_i}{\sum_{i \in R_{(j)}} w_i},$$

where replicate set  $R_{(j)}$  is defined in Section 2.2.3 and the mean estimate

$$\bar{y} = \frac{\sum_{i \in R} w_i \cdot y_i}{\sum_{i \in R} w_i}$$

is defined in Section 1.1. To employ a standardized procedure in the calculation in analysis, the replicates are computed by applying replicate weights. For details of generating replicate weights, see Appendix B. The variance of  $\bar{y}$  is estimated by

$$v_J(\bar{y}) = \frac{J-1}{J} \sum_{j=1}^J \left( \bar{y}_{(j)} - \bar{y} \right)^2.$$



Statistical theory shows that both the delete- $k$  JRR approach and the plain JRR approach, dropping one case in each replicate, yield consistent estimates (Shao & Wu, 1989). The comparison of the two approaches for the institute sample taking the 2004 fall ICT trial assessment can be found in Section 3.4.

*Estimation of the imputation errors and total variances.* For large-scale assessments such as ICT and NAEP, the results in report card format are based on plausible values, which are imputed values that resemble individual test scores and have approximately the same distribution of the characteristics of interest. Plausible values were developed as a computational approximation to obtain consistent estimates of group characteristics in assessments where individuals are administered a sample of items. The process of making use of plausible values introduces imputation errors in reporting errors as well (Little & Rubin, 1987). The imputation error should be included in the total variance.

The imputation error is estimated from repeating the procedure for each of  $M$  sets of plausible values. In practice such as in NAEP operation,  $M$  is set to 5. Let the score estimated by the  $m$ th set of plausible values be  $\bar{y}_m$ ,  $m = 1, \dots, M$ . The imputation error is estimated by

$$B = \sum_{m=1}^M \frac{(\bar{y}_m - \bar{y})^2}{M - 1}.$$

Then the total variance is estimated by

$$v_T(\bar{y}) = v_J(\bar{y}) + (1 + M^{-1}) B,$$

where  $(1 + M^{-1})$  is a finite population correction factor. The estimation process mimics that of operational NAEP: The calculation of  $v_J(\bar{y})$  is based on the first plausible value, and the estimation of  $B$  is based on all five plausible values. For details, see the “NAEP 1998 Technical Report” (Allen et al., 2001).

### 3. An Example

As a numerical example, an institute ICT sample for the fall 2004 trial assessment is used to illustrate the weighting process and variance estimation approach, including the algorithm used to aggregate students into clusters by their demographic characteristics. The base population of interest in the study consisted of 9,340 students aged 18 and above and was

stratified into two groups, freshman and junior, with the junior group including native rising and transfer rising juniors. Table C1 in Appendix C provides the information for the base population in the study. According to the sampling design, a sample of 800 students was drawn from the stratified population without replacement. Students were selected by simple random sampling within each cell. Due to case nonresponse, the assessed sample consisted of 135 freshman students and 96 junior students. Tables C2 and C3 in Appendix C provide information about the specified sample allocation and realized sample separately.

### 3.1 The Computation of Base Weights for the 2004 Example

By design, the selection probability ( $p_{hg.}$ ) for the students in group  $g$  in stratum  $h$  is approximately  $n_{hg.} / N_{hg.}$ , where  $n_{hg.}$  and  $N_{hg.}$  are the sample size and the population size of the group  $g$  in stratum  $h$ . However,  $p_{hg.}$  is not always well defined, as in the example, because sometimes institutes fail to provide necessary and accurate information of the population of interest. Using the sampling design of the 2004 fall example,  $p_{h2}$  was higher than  $p_{h1}$ . Table 1 presents the selection probabilities for the institute for the 2004 fall example.

**Table 1**

***The Selection Probability for the Cases in Each Subgroup for the 2004 Fall Example***

$p_{hg.}$		Stratum: $h$	
		Freshman	Junior
$g$	URM: No	0.063	0.075
	URM: Yes	0.255	0.178

*Note.* Underrepresented minority (URM) refers to students who are African American, Native American, Hispanic American, and Pacific Islander American.

### 3.2 Nonresponse Adjustment for the 2004 Fall Example

After the base weights are created, the weights are subject to nonresponse adjustment. The adjustment classes were formed by the variable of underrepresented minority (URM), which refers to students who are African American, Native American, Hispanic American, and Pacific Islander American, in each stratum. Table 2 provides the nonresponse adjustment factor  $f_{A,hg.}$

for each class. This factor was used to account for nonresponse of those students who were invited to the assessment but did not appear at the test.

**Table 2**

*The Nonresponse Factor for Cases in Each Class for the 2004 Fall Example*

$f_{A,hg.}$		Stratum: $h$	
		Freshman	Junior
$g$	URM: No	2.553	3.877
	URM: Yes	4.281	5.733

*Note.* Underrepresented minority (URM) refers to students who are African American, Native American, Hispanic American, and Pacific Islander American.

### 3.2 Poststratification Adjustment for the 2004 Fall Example

The adjustment cells for poststratification were formed by the variable of gender within each stratum (freshman and junior). Let the symbol  $hs$  in the subscript stand for stratum and gender. Table 3 lists the poststratification factor  $f_{hs}^r$  for each cell.

**Table 3**

*The Poststratification Factor for Cases in Each Cell for the 2004 Fall Example*

$f_{hs}^r$		Gender: $s$	
		Female	Male
$h$	Freshman	0.89037	1.15624
	Junior	1.00852	0.98210

Let the cases be aggregated by gender within each stratum. The weight for a case with  $h = 1$  and  $s = 1$  (stratum = freshman and gender = female) is adjusted by multiplying it by  $f_{11}^r$ :

$$w_{hsk} = w_{A,hs.} * f_{11}^r = w_{A,hs.} * 0.89037.$$

The case weights within a stratum were normalized to the stratum population total. The trimming procedure was not applied to case weights of the 2004 fall example because no extreme weights were found after the adjustment for nonresponse.

### ***3.3 Forming Student Clusters for the Delete- $k$ JRR for the 2004 Fall Example***

The demographic variables involved in forming clusters in variance estimation were gender and ethnicity. The cases within each stratum were first classified into four groups: (a) male and URM, (b) male and non-URM, (c) female and URM, and (d) female and non-URM. Then, cases were randomly sorted within each group. The merge-dilute algorithm was used to form the clusters utilized by the delete- $k$  JRR approach, each cluster comprising 4 students from four different groups, whenever possible. For the data of the 2004 fall example, 34 and 24 clusters were formed for the freshman and junior subsamples, respectively. To check the efficacy of the delete- $k$  JRR approach, two alternative clustering schemes also were considered. One scheme employed only gender to form clusters, each of which contained 2 male and 2 female students, if possible. The second scheme randomly chose 4 cases to form a cluster. After clusters were formed, the replicate weights were generated by a SAS program. Then, the variances were estimated by implementing the delete- $k$  JRR approach as described in Section 2.3.

### ***3.4 Empirical Results***

The numerical results for the 2004 fall example were used to compare the delete- $k$  JRR approach with the plain JRR approach. Additionally, the results were used to examine the effects of different cluster forming schemes.

First, the results showed that the delete- $k$  JRR approach provided equivalent results to those derived by the plain JRR approach. The standard error of the institute mean, estimated by the delete- $k$  JRR approach with clusters formed by the merge-dilute algorithm, was 1.51. The standard errors of means for freshmen and juniors were 2.09 and 2.19, respectively (see the first column of Table 4). By the plain JRR approach, where one case in each replicate within each stratum is dropped, the standard error of the institute mean was 1.45, and the standard errors for freshmen and juniors were 1.99 and 2.12, respectively.

Although the two JRR approaches yield estimates that are close in size, the delete- $k$  JRR approach is preferred because it provides the flexibility to analyze data from different sample designs for variant institutions by using an integrated strategy for the program. This capability is

important for institutional surveys because of the very diverse populations in institutions. Moreover, the samples for institutional surveys are usually selected by complex sampling procedures such as cluster sampling. For example, classes are often naturally used as clusters in the sampling process in institutional surveys. The delete- $k$  JRR approach uses sampled clusters in the estimation of clustering effects, but the plain JRR approach ignores possible clustering effects in complex sampling (Cochran, 1977).

Second, the results showed that different cluster forming schemes yield consistent results for the standard errors by the delete- $k$  JRR approach. In Table 4, the first column contains the standard errors from the scheme, using gender and minority status to form clusters. This scheme attempts to form clusters with homogeneous demographics. The second column has the results from the scheme using only gender to form clusters. The results in the third column were obtained by forming clusters by randomly choosing 4 cases. All three schemes employed the merge-dilute algorithm. Table 4 shows that three different schemes yielded close results, except for the estimate for URM in the freshmen group. However, this exception group had a sample size of just 32. On average, the estimates in the first column are between those in second column and the third column. The consistency across different schemes shows that the delete- $k$  JRR approach provides robust estimates for nonpercentile type statistics.

Although different schemes provided consistent results in this example, samplers often prefer to choose a scheme to form clusters with homogeneous demographics of interest, which is the scheme used to estimate the values in the first column of Table 4. The procedure for selection of suitable demographical variables for the scheme is largely based on experience and the results of previous surveys. The NAEP samples have demonstrated how to form clusters in educational surveys (Allen et al., 2001). The findings in this study are congruent with the results of other surveys (Rust, 2004).

Table C6 in Appendix C shows the mean estimates, jackknifed standard errors, imputation errors, and total standard errors for the subgroups of the 2004 fall example. The total variance equals the sampling variance plus the imputation variance; the total standard error is the square root of the total variance.

**Table 4**

*The Jackknifed Standard Errors Computed Based on Different Cluster Forming Schemes for the 2004 Fall Example*

Group	Jackknifed SE		
	4 Groups	Gender	Random
Total	1.51	1.46	1.64
Freshmen	2.09	2.06	2.32
Juniors	2.19	2.06	2.31
Male students	2.42	2.15	2.29
Female students	1.76	1.63	1.83
Male students in freshmen group	3.87	3.28	3.33
Female students in freshmen group	2.03	2.02	2.71
Male students in juniors group	2.90	2.78	3.15
Female students in juniors group	2.88	2.56	2.47
URM students	4.52	3.82	4.37
Non-URM students	1.47	1.56	1.66
URM students in freshmen group	5.04	3.67	4.00
Non-URM students in freshmen group	2.03	2.18	2.44
URM students in juniors group	7.52	6.71	7.79
Non-URM students in juniors group	2.13	2.22	2.24

*Note.* Underrepresented minority (URM) refers to students who are African American, Native American, Hispanic American, and Pacific Islander American.

#### **4. Conclusion**

This paper has introduced the weighting procedure and the delete- $k$  JRR approach that can be applied in institutional surveys such as the ICT program. The merge-dilute algorithm is proposed to form the clusters in variance estimation by the delete- $k$  JRR approach. The algorithm allows formation of clusters of required sizes and, therefore, implementation of the delete- $k$  JRR approach to variant sampling designs for diverse needs under diverse institution conditions.

Application of the weighting procedure and the delete- $k$  JRR approach to the data from the 2004 fall example yielded consistent results for several cluster forming schemes. These findings show the efficacy and applicability of the proposed framework, which can be adapted without difficulty to various situations involving similar large-scale assessments.

## References

- Allen, N., Donoghue, J., & Schoeps, T. (2001). *The NAEP 1998 technical report* (NCES 2001-509). Washington, DC: National Center for Education Statistics.
- Cochran, W. G. (1977). *Sampling techniques* (3rd ed.). New York: John Wiley & Sons.
- Deming, W. E. (1943). *Statistical adjustment of data*. New York: Wiley.
- Deming, W. E., & Stephan, F. F. (1940). On a least squares adjustment of a sampled frequency table when the expected marginal tables are known. *Annals of Mathematical Statistics*, 11, 427–444.
- Haberman, S. J. (1979). *Analysis of qualitative data* (Vol. 2). New York: Academic Press.
- Jenkins, F., & Qian, J. (2005). *IRT analysis for the ICT Fall 2004 pilot assessment*. Paper presented at the annual meeting of the National Council on Measurement in Education, Montreal, Quebec, Canada.
- Kish, L. (1965). *Survey sampling*. New York: John Wiley & Sons.
- Kish, L. (1992). Weighting for unequal Pi. *Journal of Official Statistics*, 8, 183–200.
- Knuth, D. E. (1968). *Art of computer programming*, Vol. I: Fundamental algorithms. Reading, MA: Addison-Wesley.
- Little, R. J. A., & Rubin, D. B. (1987). *Statistical analysis with missing data*. New York, Wiley.
- Lovasz, L., Pelikan, J., & Vesztergombi, K. (2003). *Discrete mathematics: Elementary and beyond*. New York: Springer-Verlag.
- Potter, F. J. (1990). A study of procedures to identify and trim extreme sampling weights. In *Proceedings of the section on survey research methods* (pp. 225–230). Alexandria, VA: American Statistical Association.
- Rust, K. F. (1985). Variance estimation for complex estimators in sample surveys. *Journal of Official Statistics*, 1, 381–397.
- Rust, K. (2004). Sampling and field operations at Westat, 1983 to 2001. In L.V. Jones & I. Olkin (Eds.), *The nation's report card: Evolution and perspectives* (pp. 427–448). Bloomington, IN: Phi Delta Kappa International.
- Shao, J., & Tu, D. (1995). *The jackknife and the bootstrap*. New York: Springer-Verlag.
- Shao, J., & Wu, C. F. J. (1989). A general theory for jackknife variance estimation. *The Annals of Statistics*, 17, 1176–1197.

- Williamson, D., Katz, I., & Redman, P. (2005). *Design and architecture of the ICT Literacy Assessment*. Paper presented at the annual meeting of the National Council on Measurement in Education, Montreal, Quebec, Canada.
- Wolter, K. (1985). *Introduction to variance estimation*. New York: Springer-Verlag.



## Notes

<sup>1</sup> Raking refers to the procedure that makes use of the Deming-Stephan algorithm to adjust weights by iterative proportional fitting. Detailed descriptions can be found in Section 1.4.

<sup>2</sup> Frame defects refer to the problems with a sampling frame such as noncoverage, undercoverage, overcoverage, duplication, or misclassification. For details, see Kish (1965).

<sup>3</sup> Suppose  $k$  is a natural number. Then for two integers  $a$  and  $c$ ,  $a$  is congruent to  $c$  modulo  $k$  [written:  $a \equiv c \pmod{k}$ ] if and only if  $a$  and  $c$  give the same remainder on division by  $k$ .

## Appendix A

### SAS Program

The SAS program implements the merge-dilute algorithm.

```
*****;
*****;
***      The SAS program implements the merge-dilute algorithm      ***;
***                                                                 ***;
*****;
*****;

**** Assign the Library Names****;
libname ict_mda 'C:\...\Merge_dilute_Algorithm';
libname ict_wts 'C:\...\Weights';
options mprint; ** mlogic symbolgen;

**** The Grouping1 file is based on Weights file from lib ict_wts ****;
**** The VARs on Weights file: case ID, case weight, group index, etc.****;
**** The VAR i_star is created for the new index in ict_mda.Grouping1 file****;
data ict_mda.Grouping1; set ict_wts.Weights;
    i_star = .;
run;
data ict_mda.Grouping; set ict_wts.Weights;
run;
**** The grouping1 is created as a temporary file in the Workspace ****;
data grouping1; set ict_mda.Grouping1;
run;

**** SAS Macro implements the merge-dilute algorithm to merge two groups ****;
**** For_Merge Macro takes three VARs as input ****;
**** VAR1: Counter, Number of Merges done by the Algorithm ****;
**** VAR2: Group1, Name/Value of the Subgroup 1 ****;
**** VAR3: Group2, Name/Value of the Subgroup 2 ****;

%macro for_merge(counter, group1, group2);

**** Dataset processing1 contains the data of group1 ****;
/* The temporary variable x is used for sorting the cases. If the new index
is not yet generated, then x is assigned a random value (<1) for those cases;
otherwise x is assigned the new index for cases in the group. The seed in
random number generator is 587. */

data processing1; set Grouping&counter.;
    if w_group = &group1.;
    if i_star = . then x = ranuni(587); else x = i_star;
run;
proc sort data = processing1; by x;
run;

**** Count the sample size of Group1 ****;
proc contents data = processing1;
    ods output Attributes = for_ct1;
run;
data for_ct1; set for_ct1;
    if _n_ = 1;
run;

**** Macro Variable of n1_dash is assigned the sample size of Group1 ****;
data _null_ ; set for_ct1;
    call symput("n1_dash",cvalue2);
```

```

run;
data processing11; set processing1;
    drop x;
    i1_dash = _n_;
run;

***** Dataset processing2 contains the data of group2 *****;
data processing2; set Grouping&counter.;
    if w group = &group2.;
    if i_star =. then x = ranuni(587); else x = i_star;
run;
proc sort data = processing2; by x;
run;
***** Count the sample size of Group2 *****;
proc contents data = processing2;
    ods output Attributes = for_ct2;
run;
data for ct2; set for_ct2;
    if _n_ = 1;
run;

***** Macro Variable of n2_dash is assigned the sample size of Group2 *****;
data _null ; set for ct2;
    call symput("n2_dash",cvalue2);
run;
data processing21; set processing2;
    drop x;
    i2_dash = _n_;
run;

***** Assign values to macro variables of k, r, k*r *****;
data _null ;
    n0_dash = mod(&n2_dash, &n1_dash);
    k = (&n2_dash - n0_dash) / &n1_dash;
    r = (&n1_dash - n0_dash);
    kr = k * r;
    call symput("n0_dash",n0_dash);
    call symput("k",k);
    call symput("r",r);
    call symput("kr", kr);
run;

***** Calculate the New Index for cases in Group1 *****;
data processing12; set processing11;
    if (i1_dash <= &r.) then i_star = (&k. + 1) * i1_dash;
    else i_star = (&k. + 2) * (i1_dash) - (&r.);
run;

***** Calculate the New Index for cases in Group2 *****;
data processing22_half; set processing21;
    alpha = i2_dash - 1;
    Beta = alpha - (&kr.);
    alpha mod k = mod(alpha, &k.);
    Beta_mod_kp1 = mod(Beta, (&k. + 1));
run;
data processing22; set processing22_half;
    if (i2_dash<=&kr.) then i_star=((&k.+1)*((alpha-
    alpha mod k)/(&k.)))+alpha mod k+1;
    else i_star=((&k.+1)*(&r.))+((&k.+2)*(Beta-
    Beta mod kp1)/(&k.+1))+Beta mod kp1+1;
    drop alpha beta alpha_mod_k Beta_mod_kp1;
run;

***** Drop the intermediate Calculation Variables *****;

```

```

data processing13; set processing12;
    drop i1_dash;
run;
data processing23; set processing22;
    drop i2_dash;
run;

***** Merge two Groups into file processing3 *****;
***** The file includes the New Index and the renewed group variable *****;
data processing3; set processing13 processing23;
    w_group = &group1. || &group2.;
run;
proc sort data = processing3; by Student_id;
run;
proc sort data = grouping&counter.; by Student_id;
run;
data _null ;
    countp = (&counter.) + 1;
    call symput("countp",left(countp));
run;
*****Create Grouping(counter+1)dataset and merge it with processing3 *****;
data Grouping&countp;
    merge grouping&counter(in=in1) processing3(in=in2);
    by Student_id;
    if(in1 = 1);
run;
***** QC the values of the macro variables *****;
%put &n1 dash;
%put &n2 dash;
%put &n0_dash;
%put &k;
%put &r;
%put &kr;

%mend for_merge;

**** To merge 4 groups, it needs to run the algorithm three times *****;
**** Assume subgroup1 and subgroup2 are two smallest groups, and merge them
****;
%for_merge(1, 'subgroup1', 'subgroup2');
**** Assume subgroup3 and subgroup1+2 are two smallest groups, and merge them *****;
%for_merge(2, 'subgroup3', 'subgroup1+2');
**** Assume subgroup4 and subgroup1+2+3 are two smallest groups, and merge them *****;
%for_merge(3, 'subgroup4', 'subgroup1+2+3');

proc sort data = ict_mda.Grouping; by Student_id;
run;
***** use Grouping COUNTER+1(4=3+1) *****;
proc sort data = Grouping4; by Student_id;
run;
***** Recover the variables (w_group, etc.) in the original file by merge *****;
data ict_mda.Grouping_with_istar;
    merge grouping4 ict_mda.Grouping;
    by Student_id;
run;
***** The sorted file Grouping_with_istar includes the New Index i_star ****;
proc sort data = ict_mda.Grouping_with_istar; by i_star;
run;

*****
*** The merge-dilute algorithm has been accomplished. Next step: ***;
*** based on required cluster size, generate the cluster index ***;
*** from the sorted file of ict_mda.Grouping_with_istar. ***;
*****

```

## Appendix B

### Computation of Replicate Weights

The replicate weights were created based on the student base weights obtained in Section 1.2. The replicate weights for the freshmen and juniors were computed separately. Each replicate was formed by dropping one of the clusters that was formed as described in Section 2.2.3. Let  $j$  be the replicate index and  $r$  be the cluster index. Let  $J_h$  be the number of clusters formed in stratum  $h$ . Each set of replicate weights,  $\{w_{hgk,r}^j\} (j = 1, \dots, J_h)$ , is defined as follows:

$$w_{hgk,r}^j = \begin{cases} w_{hgk,r}^j & r \neq j; \\ 0 & r = j. \end{cases}$$

Then, each set of replicate weights is adjusted by nonresponse adjustments and by poststratification. The procedures of nonresponse adjustments and poststratification are the same as those for the base weights, as described in Sections 1.3 and 1.4. After the adjustments, each set of replicate weights  $\{w_{hgk,r}^j\}$  is normalized to the stratum total by multiplying it by a ratio, the sum of overall case weights over the sum of the current set of replicate weights. For the institute of the 2004 fall example, 34 sets of replicate weights were generated for freshmen and 24 sets of replicate weights were generated for juniors.

To facilitate implementing the JRR approach, the replicate weights in different strata are usually assembled into one set of replicates. For the institute of the 2004 fall example, the freshman replicates and the junior replicates were assembled into a set of 34 replicates. The first 24 replicates in the set were formed by stacking the freshman replicates and junior replicates together. Because there were only 24 junior replicates, their case weights were used in the position of the last 10 replicates for the juniors.

## Appendix C

### Population Information and Sampling Plan for the 2004 Fall Example

**Inclusion Criteria:**

Age: 18+

Gender: Female or male (i.e., nonmissing)

Class: Freshman (0–44 earned credits) or “Rising Junior” (RJ) (75–104 earned credits)

Ethnicity: African American, Native American, White American, Hispanic American, Asian American, or Pacific Islander/Hawaiian American

**Table C1**

***Base Population for the 2004 Fall Example***

URM: <i>g</i>	Group			Total
	Native freshman	Native RJ	Transfer RJ	
No				
Count	4,153	2,621	1,545	8,319
% within group	88.6%	90.3%	88.5%	89.1%
Yes				
Count	537	283	201	1,021
% within group	11.4%	9.7%	11.5%	10.9%
Total	4,690	2,908	1,746	9,340

*Note.* RJ = rising junior; URM = underrepresented minority, which refers to African American, Native American, Hispanic American, and Pacific Islander American students.

**Table C2*****Sampling Design for the 2004 Fall Example (n = 800)***

URM: <i>g</i>	Group			Total
	Native freshman	Native RJ	Transfer RJ	
No				
Count	263	160	154	535
% within group	65.8%	70.5%	65.5%	66.9%
Yes				
Count	137	40	46	265
% within group	34.3%	29.5%	34.5%	33.1%
Total	400	200	200	800

*Note.* RJ = rising junior; URM = underrepresented minority, which refers to African American, Native American, Hispanic American, and Pacific Islander American students.

**Table C3*****Unweighted Counts of the Assessed Sample***

URM: <i>g</i>	Freshman	Junior	Total
No			
Count	103	81	184
% within group	76.30%	84.38%	79.65%
Yes			
Count	32	15	47
% within group	23.70%	15.63%	20.35%
Total	135	96	231

*Note.* URM = underrepresented minority, which refers to African American, Native American, Hispanic American, and Pacific Islander American students.

**Table C4*****Weighted Counts Without Adjustment for the 2004 Fall Example***

URM: g	Freshman	Junior	Total
No			
Count	4,173.96	4,187.16	8,361.12
% within group	88.60%	89.66%	89.12%
Yes			
Count	537.21	483.12	1,020.33
% within group	11.40%	10.34%	10.88%
Total	4,711.17	4,670.28	9,381.45

*Note.* URM = underrepresented minority, which refers to African American, Native American, Hispanic American, and Pacific Islander American students.

**Table C5*****Weighted Counts With Adjustment for the 2004 Fall Example***

URM: g	Freshman	Junior	Total
No			
Count	4,134.72	4,188.38	8,323.10
% within group	88.62%	89.60%	89.11%
Yes			
Count	530.73	486.17	1,016.90
% within group	11.38%	10.40%	10.89%
Total	4,665.45	4,674.55	9,340.00

*Note.* URM = underrepresented minority, which refers to African American, Native American, Hispanic American, and Pacific Islander American students.



**Table C6**

*Mean Estimates, Jackknifed Standard Errors, Imputation Errors, and Total Standard Errors for the 2004 Fall Example*

Group	Jackknifed SE					Total SE		
	Mean	4 Groups	Gender	Random	Imputation error	4 Groups	Gender	Random
Total	172.75	1.51	1.46	1.64	0.84	1.73	1.68	1.84
Freshmen	171.28	2.09	2.06	2.32	1.53	2.59	2.57	2.78
Juniors	172.39	2.19	2.06	2.31	0.95	2.39	2.27	2.50
Male students	176.98	2.42	2.15	2.29	1.24	2.72	2.48	2.61
Female students	168.71	1.76	1.63	1.83	1.27	2.17	2.07	2.23
Male students in freshmen group	175.78	3.87	3.28	3.33	2.78	4.76	4.30	4.34
Female students in freshmen group	170.97	2.03	2.02	2.71	0.64	2.13	2.12	2.79
Male students in juniors group	178.18	2.90	2.78	3.15	1.59	3.31	3.20	3.53
Female students in juniors group	166.44	2.88	2.56	2.47	2.19	3.62	3.37	3.30
URM	167.31	4.52	3.82	4.37	0.98	4.63	3.94	4.48
Non-URM	173.40	1.47	1.56	1.66	0.97	1.76	1.83	1.92
URM in freshmen group	170.21	5.04	3.67	4.00	3.61	6.20	5.15	5.38
Non-URM in freshmen group	173.37	2.03	2.18	2.44	1.77	2.70	2.81	3.01
URM in juniors group	164.39	7.52	6.71	7.79	2.79	8.02	7.27	8.28
Non-URM in juniors group	173.44	2.13	2.22	2.24	1.06	2.38	2.46	2.48

*Note.* The results listed in the table are used to explain the methodologies used instead of reporting. The ICT program uses the minimum sample size of 50 as the standard for reporting significance tests or for generalizing to the campus population. URM = underrepresented minority, which refers to African American, Native American, Hispanic American, and Pacific Islander American students.